

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 10011338

(43)Date of publication of application: 16.01.1998

(51)Int.Cl.

G06F 12/00

G06F 12/00

(21)Application number: 08164008

(71)Applicant:

HITACHI INF SYST LTD

(22)Date of filing: 25.06.1996

(72)Inventor:

IKEDA KUNISAKU

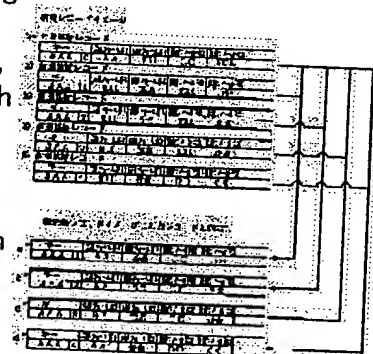
YAGI HIROAKI

(54) RELATIONAL DATABASE SYSTEM, METHOD FOR STORING AND READING DATA IN IT AND RECORDING MEDIUM FOR RECORDING PROGRAM FOR IT

(57)Abstract:

PROBLEM TO BE SOLVED: To permit the reduction of storing capacitance and the improvement of access efficiency to be compatible by providing a record reading and restoring means, etc., for synthesizing an applying common storing record which is read from a main key and a sub-key with an individual storing record into one record and for restoring an input record.

SOLUTION: An after restoration record a' extracts information 'AA' and 'CC' of common fields 1 and 2 in the common storing record (1) as information of the common fields 1 and 2 and also extracts information 'aa' and 'ii' of the individual fields 1 and 2 in the individual storing record (2) as information of the individual fields 1 and 2 so as to execute synthesization. The after restoration record b' extracts information 'AA' and 'CC' of the common fields 1 and 2 in the common storing record (1) as information of the common fields 1 and 2 and also extracts information 'uu' and 'ee' of the individual fields 1 and 2 in the individual storing record (3) as information of the individual fields 1 and 2 so as to execute synthesization.



(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 1 2		G 0 6 F 12/00	5 1 2
	5 1 1			5 1 1

審査請求 未請求 請求項の数6 O L (全 14 頁)

(21) 出願番号 特願平8-164008

(22) 出願日 平成8年(1996) 6月25日

(71) 出願人 000152985

株式会社日立情報システムズ

東京都渋谷区道玄坂1丁目16番5号

(72) 発明者 池田 国作

東京都渋谷区道玄坂一丁目16番5号 株式会社日立情報システムズ内

(72) 発明者 八木 宏明

東京都渋谷区道玄坂一丁目16番5号 株式会社日立情報システムズ内

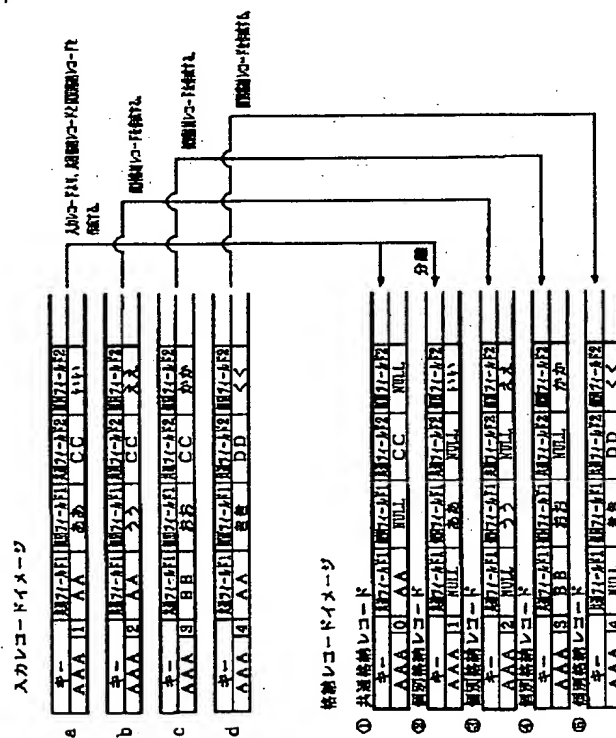
(74) 代理人 弁理士 磯村 雅俊 (外1名)

(54) 【発明の名称】 リレーショナル・データベース・システム、該システムへのデータ格納・読み出し方法、およびそのためのプログラムを記録した記録媒体

(57) 【要約】

【課題】 格納容量の削減とアクセス効率向上を両立させることが可能なRDB、該RDBへのデータ圧縮格納／検索時の読み出し復元を行うデータ格納・読み出し方法、およびそのための制御プログラムを記録した記録媒体を提供する。

【解決手段】 格納データレコード(入力レコードa～d)を、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部同値の複数レコードを、主キー部と共通データフィールド部とNULL値フィールド部とで構成される一つの親レコード(共通格納レコード①)と、主キー部と、サブキー部と、個別データフィールド部とNULL値フィールド部とで構成される複数の子レコード(個別格納レコード②～⑤)とに分割してデータベースに格納する。また、指定された主キーとサブキーから読み出した該当の親レコードと、子レコードとを1レコードに合成し、格納データレコードを復元する。



【特許請求の範囲】

【請求項1】 リレーショナル・データベース・システムにおいて、格納データレコードを、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部が同値を持つデータレコードが複数レコード存在する場合に、

主キー部同値の複数レコードを、主キー部と共通データフィールド部とNULL値フィールド部とで構成される一つの親レコード（共通格納レコード）と、主キー部と、サブキー部と、個別データフィールド部とNULL値フィールド部とで構成される複数の子レコード（個別格納レコード）とに、分割して格納するレコード分割・格納手段と、

指定された主キーとサブキーから読み出した該当の親レコードと、子レコードとを1レコードに合成し、格納データレコードを復元するレコード読み出し復元手段とを備えたことを特徴とするリレーショナル・データベース・システム。

【請求項2】 請求項1記載のリレーショナル・データベース・システムにおいて、

親レコードと、前記親レコードに対応する子レコードを、共に同じレコード・フォーマット、レコード長で作成し、親レコードにおける共通データフィールド部は、子レコードにおけるNULL値フィールド部に、また、子レコードにおける個別データフィールド部は、親レコードにおけるNULL値フィールド部に、それぞれ対応するように格納データレコードをフォーマットすることを特徴とするレコード分割、格納手段を有するリレーショナル・データベース・システム。

【請求項3】 リレーショナル・データベースにおけるデータレコードの格納・読み出し方法において、

格納データレコードを、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部が同値を持つデータレコードが複数レコード存在する場合に、主キー部同値の複数レコードを、主キー部と共通データフィールド部とNULL値フィールド部とで構成される一つの親レコードと、主キー部と、サブキー部と、個別データフィールド部とNULL値フィールド部とで構成される複数の子レコードとに分割してデータベースに格納し、指定された主キーとサブキーから読み出した該当の親レコードと、子レコードとを1レコードに合成し、格納データレコードを復元するようにしたデータレコード格納・読み出し方法。

【請求項4】 請求項3記載のデータレコード格納・読み出し方法において、親レコードと、前記親レコードに対応する子レコードを、共に同じレコード・フォーマット、レコード長で作成し、親レコードにおける共通データフィールド部は、子レコードにおけるNULL値フィールド部に、また、子レコードにおける個別データフィールド部は、親レコードにおけるNULL値フィールド

部に、それぞれ対応するよう格納データレコードをフォーマットすることを特徴とするデータレコード格納・読み出し方法。

【請求項5】 リレーショナル・データベース・システムにおけるデータベース・アクセス制御プログラムであって、格納データレコードを、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部が同値を持つデータレコードが複数レコード存在する場合に、主キー部同値の複数レコードを、主キー部と、共通データフィールド部と、NULL値フィールド部とで構成される一つの親レコードと、主キー部と、サブキー部と、個別データフィールド部と、NULL値フィールド部とで構成される複数の子レコードとに、分割して格納するレコード分割・格納手順と、指定された主キーとサブキーから読み出した該当の親レコードと、子レコードとを1レコードに合成し、格納データレコードを復元するレコード読み出し復元手順を含んだデータベース・アクセス制御プログラムが記録された記録媒体。

【請求項6】 請求項5記載の記録媒体において、前記レコード分割・格納手順は、親レコードと、前記親レコードに対応する子レコードを、共に同じレコード・フォーマット、レコード長で作成し、親レコードにおける共通データフィールド部は、子レコードにおけるNULL値フィールド部に、また、子レコードにおける個別データフィールド部は、親レコードにおけるNULL値フィールド部に、それぞれ対応するよう格納データレコードをフォーマットするものであることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、リレーショナル・データベース・システム（Relational Data Base System：以下RDBと略す）に関するものであり、特に、データを大幅に圧縮して格納することが可能なRDB、該RDBへのデータ圧縮格納および読み出し復元を行うデータ格納・読み出し方法、およびそのための制御プログラムを記録した記録媒体に関する。

【0002】

【従来の技術】RDBは、データを複数の表の形式で記憶するデータベースである。各表はデータ要素の集まりである行（レコードともいう）と列から構成され、各表は他の表と区別するための表名を持っている。RDBにおける各表の行の順序は任意であり、また、一つ以上の表に対して自由に選択（Selection：表から特定の条件に合致した行を取り出す／Restriction：制約）、射影（Projection：表から特定の列だけを取り出す）、結合（Join：2つ以上の表から1つの表を生成）などの関係演算、合併（Union：2つの照会結果を接続して合わせる）、共通（Intersection：複数の表間で共通の

内容を取り出す)、差(Difference:複数の表間の差分データを取り出す)などの集合演算、挿入(Insert:行単位での挿入)、削除>Delete:行単位での削除)、更新(Update:データ要素単位での更新)などの変更操作が可能である。RDBは、表表現であるため一覧性があり親しみやすくまた各種操作が理解しやすいために広く研究されて様々な改良が加えられている。なお、RDBの詳細については、例えば、平尾隆行著「関係データベースシステム」(1991年3月20日 近代科学社発行)を参照されたい。

【0003】RDBの表定義設計は、対象業務の論理データモデルを設計し、正規化手順を経てRDBの表定義における列定義を決定する。通常、正規化手順により表列定義の主キー以外の項目属性の間に従属関係がなくなるまで正規化を行い、各表の列定義を決定する。また、正規化を行うことで、各表の列同志の項目重複を避けることができる。以下、その例を具体的に説明する。

【0004】(例題)ある契約マスタの非正規化型として論理データ項目が以下のケースを想定する。

1. 証券番号
2. 顧客番号
3. 契約者氏名
4. 契約者住所
5. 契約者電話番号
6. 証券種類
7. 契約明細項目1(ex:契約期間)
8. 契約明細項目2(ex:契約料)

【0005】これらのデータにおいて、同一顧客が契約するセット商品の契約情報をRDB上で保有管理する場合、個別の契約内容を(証券番号+枝番)で登録管理するケースでは、表定義の列定義項目として以下の構成が考えられる。

(表定義A)

1. 証券番号
2. 証券番号枝番
3. 顧客番号
4. 契約者氏名
5. 契約者住所
6. 契約者電話番号
7. 証券種類
8. 契約明細項目1(ex:契約期間)
9. 契約明細項目2(ex:契約料)

【0006】キーとして「証券番号+証券番号枝番」を考えた場合、表定義Aにおいてデータ項目3~6については、顧客番号をキーとする従属関係が存在するため、正規化によりRDBにおけるデータモデルの表定義は、次のように2つに分割される。

(表定義1)

1. 証券番号
2. 証券番号枝番

3. 顧客番号
 4. 証券種類
 5. 契約明細項目1(ex:契約期間)
 6. 契約明細項目2(ex:契約料)
- (表定義2)

1. 顧客番号
2. 契約者氏名
3. 契約者住所
4. 契約者電話番号

10 【0007】上記のような表定義においてマスタ内容の照会処理を考えた場合、表へのアクセスは、

1. 表定義1からの契約明細の検索アクセス
 2. 表定義2から該当契約の契約者情報(氏名、住所、電話番号など)の検索アクセス
- の2回の表アクセスを行うことになる。

【0008】

【発明が解決しようとする課題】上記例題の表定義構成において、次のような条件を有する場合には、表へのアクセス時間が問題となってくる。

- 20 1. 主キーの枝番で保管する同一顧客の契約レコードが平均2件以上存在する。(同一キーで検索できる表レコードが2件以上存在する)
2. 保有契約件数が多数存在する。
3. 表の1レコード内に従属関係の項目が多数存在する。

(上記例題での契約者氏名、契約者住所、契約者電話番号など)

【0009】上記のような条件のデータ項目を保有する表定義を考えた場合、格納容量を削減するためには、上記例題のように、従属関係項目を別表定義として分離し、2表にすることが考えられる。しかしながら、このようにした場合には格納容量は削減できるが、アクセス対象の表が2表になるため、アクセス効率が低下するという新たな問題を招く。また、全てのデータ項目を1表で定義した場合、表へのアクセスは、1表のみとなり表へのアクセス効率は向上するが、表列定義内の各レコードで同一主キー同志で「同一な従属関係の項目」が多数重複して存在するため、RDBの格納容量が膨大となる。従って、上記のようなケースでは、格納容量の削減とアクセス効率向上を両立させることが困難になる。本発明は、上記問題点を解決し、格納容量の削減とアクセス効率向上を両立させることが可能なRDB、該RDBへのデータ圧縮格納/検索時の読み出し復元を行うデータ格納・読み出し方法、およびそのための制御プログラムを記録した記録媒体を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明は、上記目的を達成するために、RDBにおいて、格納データレコード

(入力レコードa~d)を、主キー部と、サブキー部

50 と、共通フィールド部と、個別フィールド部とに区分け

し、主キー部が同値を持つデータレコードが複数レコード存在する場合に、主キー部同値の複数レコードを、主キー部と共通データフィールド部とNULL値フィールド部とで構成される一つの親レコード（共通格納レコード①）と、主キー部と、サブキー部と、個別データフィールド部とNULL値フィールド部とで構成される複数の子レコード（個別格納レコード②～⑤）とに、分割して格納するレコード分割・格納手段と、指定された主キーとサブキーから読み出した該当の親レコード（共通格納レコード①）と、子レコード（個別格納レコード②～⑤）とを1レコードに合成し、格納データレコード（入力レコードa～d）を復元するレコード読み出し復元手段とを備えたことを特徴としている。

【0011】また、この場合、親レコード（共通格納レコード①）と、前記親レコード（共通格納レコード①）に対応する子レコード（個別格納レコード②～⑤）を、共に同じレコード・フォーマット、レコード長で作成し、親レコード（共通格納レコード①）における共通データフィールド部は、子レコード（個別格納レコード②～⑤）におけるNULL値フィールド部に、また、子レコード（個別格納レコード②～⑤）における個別データフィールド部は、親レコード（共通格納レコード①）におけるNULL値フィールド部に、それぞれ対応するように格納データレコード（入力レコードa～d）をフォーマットすることを特徴としている。

【0012】また、本発明のリレーショナル・データベースにおけるデータレコードの格納・読み出し方法は、格納データレコード（入力レコードa～d）を、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部が同値を持つデータレコードが複数レコード存在する場合に、主キー部同値の複数レコードを、主キー部と共通データフィールド部とNULL値フィールド部とで構成される一つの親レコード（共通格納レコード①）と、主キー部と、サブキー部と、個別データフィールド部とNULL値フィールド部とで構成される複数の子レコード（個別格納レコード②～⑤）とに分割してデータベースに格納し、指定された主キーとサブキーから読み出した該当の親レコード（共通格納レコード①）と、子レコード（個別格納レコード②～⑤）のいずれかとを1レコードに合成し、格納データレコード（入力レコードa～d）を復元することを特徴としている。

【0013】また、この場合、親レコード（共通格納レコード①）と、前記親レコード（共通格納レコード①）に対応する子レコード（個別格納レコード②～⑤）を、共に同じレコード・フォーマット、レコード長で作成し、親レコード（共通格納レコード①）における共通データフィールド部は、子レコード（個別格納レコード②～⑤）におけるNULL値フィールド部に、また、子レコード（個別格納レコード②～⑤）における個別データ

フィールド部は、親レコード（共通格納レコード①）におけるNULL値フィールド部に、それぞれ対応するよう格納データレコード（入力レコードa～d）をフォーマットすることを特徴としている。

【0014】さらに、本発明の記録媒体は、格納データレコード（入力レコードa～d）を、主キー部と、サブキー部と、共通フィールド部と、個別フィールド部とに区分けし、主キー部が同値を持つデータレコードが複数レコード存在する場合に、主キー部同値の複数レコードを、主キー部と、共通データフィールド部と、NULL値フィールド部とで構成される一つの親レコード（共通格納レコード①）と、主キー部と、サブキー部と、個別データフィールド部と、NULL値フィールド部とで構成される複数の子レコード（個別格納レコード②～⑤）とに、分割して格納するレコード分割・格納手段と、指定された主キーとサブキーから読み出した該当の親レコード（共通格納レコード①）と、子レコード（個別格納レコード②～⑤）とを1レコードに合成し、格納データレコード（入力レコードa～d）を復元するレコード読み出し復元手段を含んだデータベース・アクセス制御プログラムが記録されたものであることを特徴としている。

【0015】また、この場合、レコード分割・格納手段は、親レコード（共通格納レコード①）と、前記親レコード（共通格納レコード①）に対応する子レコード（個別格納レコード②～⑤）を、共に同じレコード・フォーマット、レコード長で作成し、親レコード（共通格納レコード①）における共通データフィールド部は、子レコード（個別格納レコード②～⑤）におけるNULL値フィールド部に、また、子レコード（個別格納レコード②～⑤）における個別データフィールド部は、親レコード（共通格納レコード①）におけるNULL値フィールド部に、それぞれ対応するよう格納データレコード（入力レコードa～d）をフォーマットするようにしたものであることを特徴としている。

【0016】

【発明の実施の形態】表定義上で、1表にて定義した場合（表定義Aのケース）、1表の列定義項目内で従属関係にある項目については、同一主キー（枝番なし）同志のレコード間で項目値が重複して格納されてしまうため、RDB格納容量が増大する。このため、同一主キー（枝番なし）同志のレコード間での従属関係にある項目の格納を同一主キーのレコード件数分格納せず、1レコード分の格納に抑えて格納できれば、2表に分割して格納するケースとほぼ同等の格納容量となる。本発明では、以下の格納方法によりアクセス効率のよい1表による表定義RDBを使用し、レコード間の「従属関係にある項目」の重複格納を避けることにより、格納容量の削減も実現している。

【0017】（1）格納する主キーが重複するレコード

の各項目を、各レコード個別フィールド群と共通フィールド群とに振り分けて登録する。各レコード個別項目は、個別格納レコード「主キー+枝番(01~)」に格納する。個別格納レコード内の共通フィールドには、NULL値設定として格納する。主キーが重複するレコードの共通項目は、「主キー+枝番(00)」の共通格納レコードとして格納する。共通格納レコードの個別フィールドには、個別格納レコードと反対にNULL値を設定して格納する。入力レコードから共通格納レコードおよび個別格納レコードに分離するためには、RDBの列定義での各項目の内、どの項目が共通フィールドでどの項目が個別フィールドかを識別する情報が必要になる。本実施例では、本識別する情報としてMASKテーブルを定義しておき、該MASKテーブルとの突き合わせにより格納レコードへの分離を行う。

(2) レコードの復元時は、上記分割レコードを統合し、格納時のレコードを復元する。

【0018】上記発明の実施の形態によると、同一キー内(枝番などで複数のレコードが存在する)の複数レコード間で、重複している情報を、共通格納レコードとして集約するとともに、個別格納レコードは共通格納レコードと逆の重複しない情報を設定し作成している。これにより、データベースの標準機能である「連続文字圧縮機能」が有効に働き、ディスク容量を削減することができる。このとき、共通格納レコードに設定するフィールドと、個別格納フィールドに設定するフィールドの識別は、MASKテーブルとの比較によって実現させており、1項目単位に比較する方法に比較して処理効率が向上する。また、格納レコードの復元は、共通格納レコードと個別格納レコードの合成により実現させている。

【0019】以下、本発明の実施例を図面を用いて詳細に説明する。図1は本発明の実施例に係るリレーショナル・データベース1とユーザプログラム2とデータベース管理システム(DBMS)3とDBアクセス制御ルーチン4の関係を示す概念図である。DBアクセス制御ルーチン4は、ユーザプログラムから受け取った入力レコードからリレーショナル・データベース1へ格納する格納レコードを作成するとともに、リレーショナル・データベース1に格納されている格納レコードから元の入力レコードを復元するためのルーチンである。以下の説明で明らかになるように、このDBアクセス制御ルーチン4は、ユーザプログラムがデータベースに対しレコードの入出力要求を行った場合、次のような処理を実施する。

【0020】(1) レコード検索要求の場合、データベースから共通格納レコードと個別格納レコードを入力し、共通格納レコードと個別格納レコードでレコードを復元処理を行い、復元したレコードをユーザプログラムへ返す。

(2) レコード更新要求の場合、データベースから共通

格納レコードを入力し、ユーザプログラムから受け取ったレコードと該共通格納レコードとで個別格納レコードを作成し、個別格納レコードをデータベースに格納する。本発明は、このDBアクセス制御ルーチン4の機能により、ユーザプログラム2側からもデータベース管理システム(DBMS)3側からも、相互の関連を意識することなく、リレーショナル・データベース1のアクセスを可能としている。

【0021】図2は、ユーザプログラム2から入力される入力レコードからリレーショナル・データベース1へ格納する格納レコードを作成する方法を示すイメージ図である。本方法では、同一キー(枝番などで区別して管理されている同一主キーを有するデータは同一キーとして取り扱う)を有する複数の入力レコードから、共通フィールド(複数のレコード間で同一の情報が設定されている比率が大きい項目の集まり)に該同一の情報を設定し、個別フィールド(複数のレコード間で同一の情報が設定されている比率が小さい項目の集まり)の情報は「NULL値」を設定した共通格納レコードと、個別フィールド情報として対応する入力レコードの個別フィールドの値を設定し、共通フィールドの情報として「NULL値」を設定(例外的に、入力レコードの共通フィールドが共通格納レコードの共通フィールドの情報と異なる場合には入力レコードの共通フィールドの情報を優先して設定)した個別格納レコードを作成する。このようにして、共通格納レコードが同一キー毎に1レコード作成され、個別格納レコードが入力レコード数だけ作成される。

【0022】次に、図2を用いて、入力レコードから格納レコードを作成する方法をさらに詳細に説明する。図2の例では、入力レコードa~dはキーが全て同一のAAで枝番がそれぞれ1~4を有している。入力レコードa, b, dの共通フィールド1が同一の情報「AA」を有し、入力レコードcの共通フィールド1だけが情報「BB」を有している。また、入力レコードa, b, cの共通フィールド2が同一の情報「CC」を有し、入力レコードdの共通フィールド2だけが情報「DD」を有している。個別フィールド1および2の情報は全ての入力レコードで異なっている。

【0023】なお、以下の説明では、簡単のために入力レコードが4個の場合を説明しているが入力レコード数は任意でよく、多ければ多いほど本発明の効果が顕著になる。また、入力レコードの共通フィールドと個別フィールドとしてそれぞれ2個のみを示しているが、この数も任意でよいことはいふまでもない。

【0024】上述したように、共通格納レコード①は、共通フィールド1に入力レコードa, b, dで共通の情報「AA」を、共通フィールド2に入力レコードa, b, cで共通の情報「CC」を入れ、個別フィールド1および2には全て「NULL値」を入れることにより、

入力レコードa～dに対してただ一つ作成される。

【0025】個別格納レコード②～⑤の各々は、個別フィールド1および2には対応する入力レコードa～dの個別フィールド1および2の情報をそのまま入れ、共通フィールド1および2には原則として「NULL値」を入れて作成する。例外的に、入力レコードの共通フィールドの情報が共通格納レコード①の共通フィールドの情報と異なる場合に、その入力レコードに対応する個別格納レコードの共通フィールドにその入力レコードの共通フィールドの情報を優先的に設定するようにする。入力レコードa～dに上記操作を行うことにより、図2に示されているような格納レコードが作成される。特に、個別格納レコード④の共通フィールド1には「BB」が、個別格納レコード⑤の共通フィールド2には「DD」が入れられることに留意されたい。

【0026】以上の説明から明らかなように、格納レコードを作成する場合、共通フィールドの情報が1番目の入力レコードの共通フィールドの情報と同一の情報を有する入力レコードに対応する個別格納レコードの共通フィールドは「NULL値」に設定されるため、データベースの標準機能である「連続文字圧縮機能」が有効に働きデータベース格納容量を削減することができる。このことは、入力レコード数が多く、かつ入力レコードの共通フィールドができるだけ共通の情報を有しているほどデータベース格納容量（ディスク容量）の削減効果が大きいことを意味している。

【0027】なお、1番目の入力レコードの共通フィールドだけが後続する他の入力レコードの共通フィールドの値と異なっている場合には個別格納レコードの共通フィールドを「NULL値」に設定することができないため、上述した連続文字圧縮機能によるデータベース格納容量の削減効果はなくなるが、「1番目の入力レコードの共通フィールドだけが後続する他の入力レコードの共通フィールドの値と異なっている場合」という状況は確率的に小さく、ほとんどの場合に対して上記データベース格納容量（ディスク容量）の削減効果が期待できる。

【0028】次に、格納レコードから入力レコードを復元する方法について説明する。図3は、リレーショナルデータベース1に格納されている共通格納レコード①と個別格納レコード②～⑤から構成される格納レコードから入力レコードa～d（復元後レコードa'～d'として示す）を復元する方法を示すイメージ図である。レコードの復元は、共通格納レコードの共通フィールドの情報と、個別格納レコードの個別フィールドの情報を合成して行われる。ただし、個別格納レコードの共通フィールドに「NULL値」以外の情報が設定されている場合には、この個別格納レコードの共通フィールド情報を優先して採用するようにする。

【0029】図3を用いて具体的に説明すると、復元後レコードa'は、共通フィールド1および2の情報とし

て共通格納レコード①の共通フィールド1および2の情報「AA」「CC」を、個別フィールド1および2の情報として個別格納レコード②の個別フィールド1および2の情報「ああ」「いい」を抽出して合成し、復元後レコードb'は、共通フィールド1および2の情報として共通格納レコード①の共通フィールド1および2の情報「AA」「CC」を、個別フィールド1および2の情報として個別格納レコード③の個別フィールド1および2の情報「うう」「ええ」を抽出して合成する。

【0030】復元後レコードc'は、共通フィールド1の情報としては、個別格納レコード④の共通フィールド1に共通格納レコード①の共通フィールド1の情報「AA」とは異なる情報「BB」が設定されているのでこの情報「BB」を採用するとともに、共通フィールド2の情報として共通格納レコード①の共通フィールド2の情報「CC」を、個別フィールド1および2の情報として個別格納レコード④の個別フィールド1および2の情報「おお」「かか」を抽出して合成する。

【0031】復元後レコードd'は、共通フィールド2の情報としては、個別格納レコード⑤の共通フィールド2に共通格納レコード①の共通フィールド2の情報「CC」とは異なる情報「DD」が設定されているのでこの情報「DD」を採用するとともに、共通フィールド1の情報として共通格納レコード①の共通フィールド1の情報「AA」を、個別フィールド1および2の情報として個別格納レコード④の個別フィールド1および2の情報「きき」「くく」を抽出して合成する。

【0032】図2に示したように、格納レコード作成として、

(イ) 1番目の入力レコードから共通格納レコードと1番目の個別格納レコードを作成

(ロ) 2番目以降の入力レコードから2番目の個別格納レコードを作成（共通フィールドの情報が1番目の共通フィールドの情報と同一の場合）

(ハ) 2番目以降の入力レコードから2番目の個別格納レコードを作成（共通フィールドの情報が1番目の共通フィールドの情報と異なっている場合）

の3種類ある。

【0033】図4～図6は、これら3種類の格納レコード作成手順の具体例を説明するための図であり、MASKテーブルを利用している点に特徴がある。図4は同一キー（AAA）を有する入力レコードに対応する格納レコードを作成する場合の初回の格納レコード作成手順を示す図であり、この手順により1番目の入力レコードaから共通格納レコード①と1番目の個別格納レコード②が作成される。同図に示すように、初回の格納レコード作成手順は、まず、MASKテーブル1（共通フィールドがHIGH-VALUE値で、個別フィールドが「NULL値」のテーブル）と1番目の入力レコードaとで論理積をとり、共通格納レコード①を作成する。これは

論理積のビット操作により、共通格納レコード①の共通フィールドの情報は入力レコードの共通フィールドの情報のままとし、個別フィールドの情報を全て「NULL値」として作成することに相当している。

【0034】次に、MASKテーブル2（共通フィールドが「NULL値」で、個別フィールドがHIGH-VALUE値のテーブル）と入力レコードaとで論理積をとり、個別格納レコード②（1番目の入力レコードaに対応）を作成する。これは論理積のビット操作により、個別格納レコード②の共通フィールドの情報を「NULL値」とし、個別フィールドの情報は入力レコードaの情報のまま「ああ」「いい」として作成することに相当している。なお、上述したMASKテーブル1およびMASKテーブル2は、レコードコピーライブラリにより自動生成される。

【0035】図5は、同一キーを有する2番目以降の入力レコードに対応する格納レコードを作成する場合、該2番目以降の入力レコードの共通フィールドの内容が共通格納レコードの共通フィールドの内容と同一の場合（本実施例では2番目の入力レコードbが該当している）の格納レコード作成手順を説明するための図であり、同図は2番目の入力レコードbから2番目の個別格納レコード③を作成する場合の例を示している。すなわち、上記図4に示した初回の格納レコード作成手順で作成した共通格納レコード①と入力レコードbとで排他的論理和をとり、第2の個別格納レコード③を作成している。これは排他的論理和のビット操作により、個別格納レコード③の共通フィールドの情報を「NULL値」とし、個別フィールド1および2の情報は入力レコードの個別フィールドの情報のまま「うう」「ええ」として作成することを意味している。

【0036】図6は、2番目以降の入力レコードで共通フィールドの情報が共通格納レコードの共通フィールドの内容と異なる場合（本実施例では3番目の入力レコードcおよび4番目の入力レコードdが該当している）の格納レコード作成方法を説明するための図であり、これにより上記実施例の3番目の個別格納レコード④および4番目の個別格納レコード⑤が作成される。図6は3番目の入力レコードcから3番目の個別格納レコード④を作成する場合の例を示したものであるが、4番目の入力レコードdから4番目の個別格納レコード⑤を作成する場合も同様な方法がとられる。

【0037】図6の方法は図5に示した2番目以降の格納レコード作成方法と類似しているが、共通格納レコード①と入力レコードとで、排他的論理和をとったとき、共通フィールドの情報が「NULL値」にならない場合（すなわち共通フィールドの内容が共通の情報でない場合）、入力レコードの情報を優先して設定するようにしたところが図5と異なっている。このようにすることによって、入力レコードの共通フィールドの情報が全て同

一でない場合に対しても柔軟に対応できるようになる。

【0038】図5および図6の方法を実施するために、共通格納レコード①の共通フィールド（1番目の入力レコードの共通フィールドと同じ）と2番目以降の入力レコードの共通フィールドの情報が全て一致しているかどうか、また、もし不一致の場合にはどの共通フィールドの情報が不一致なのかを判別する必要がある（図6に示のように、不一致の共通フィールドに入力レコードの情報を優先して設定するため）。次に、この判別方法を図7および図8を用いて説明する。

【0039】まず、入力レコードとMASKテーブル1（共通フィールドがHIGH-VALUE値で、個別フィールドが「NULL値」のテーブル）とで論理積をとって結果1を得、次に、その結果1と共通格納レコード①とで排他的論理和をとって結果2を得る。結果2の共通フィールドと個別フィールドを含めた全フィールドが「NULL値」の場合、共通フィールド情報の値は一致していると判断する。図7にその具体例を示す。同図は、2番目の入力レコードbに対してこの処理を行った例を示したもので、結果2の全てのフィールドが「NULL値」になるので入力レコード2と共通格納レコード①の共通フィールドの情報が全て一致（共通フィールド1が「AA」、共通フィールド2が「CC」で一致）しているものと判断される。

【0040】また、結果2の全フィールドが「NULL値」でない場合は、判定する範囲を2分割しそれぞれの範囲に対して同様の処理を行い、2分割された範囲内で全フィールドが「NULL値」であれば、その範囲内では共通フィールドが一致していると判断し、全フィールドが「NULL値」でない範囲についてのみ再度上記と同様の処理を行う、という処理を繰り返し、範囲を順次絞り込んで不一致の情報を有する共通フィールドを特定する。

【0041】図8にその具体例を示す。同図は、3番目の入力レコードcに対してこの処理を行った例を示したもので、結果1の共通フィールド1の情報が「BB」で、共通格納レコード①の共通フィールド1の情報が「AA」であるので結果2の共通フィールド1の情報は「NULL値」にならない。この場合、不一致のフィールドを見付けるために、不一致である全フィールドの範囲（イ）を範囲（ロ）と範囲（ハ）に2分割して各範囲で同様の処理を行うと範囲（ハ）内の全フィールドは「NULL値」であるためこの範囲内では共通フィールドは一致している。一方、範囲（ロ）内の全フィールドが「NULL値」でないためこの範囲（ロ）内のフィールドに不一致フィールドがあることがわかる。さらに範囲（ロ）を範囲（ニ）と範囲（ホ）に2分割しそれぞれの範囲に同様の処理を行う。その結果、範囲（ホ）のフィールドは「NULL値」であるが範囲（ニ）のフィールドは「NULL値」でないため、最終的に範囲

(二)、すなわち共通フィールド1の情報が不一致であることがわかる。

【0042】図9～図11は、個別格納レコードを入力レコードイメージへ復元する方法を説明するための図である。まず、図9に示すように、個別格納レコードとMASKテーブル1（共通フィールドがHIGH-VALUE値で、個別フィールドが「NULL値」のテーブル）とで論理積をとって結果1を得る。得られた結果1の全フィールドが「NULL値」の場合、共通フィールドの情報が共通格納レコードの情報と一致していると判断できるため、個別格納レコードと共通格納レコードとで論理和をとれば、入力レコードの復元が可能となる。

【0043】例えば、図9および図10は個別格納レコード②と共通格納レコード①から入力レコードa'を復元する例であり、この場合、結果1の全フィールドが「NULL値」であるため復元する入力レコードの共通フィールドの情報が共通格納レコードの共通フィールドの情報「AA」「CC」と一致していると判定され、図10に示すように、個別格納レコード②と共通格納レコード①との論理和をとることによって復元される。すなわち、個別格納レコード②の共通フィールド1および2と共通格納レコード①の個別フィールド1および2はともに「NULL値」であるので、復元後の入力レコードa'は、共通フィールド1および2に共通格納レコード①の共通フィールド1および2の情報「AA」および「CC」が、個別フィールド1および2に個別格納レコード②の個別フィールド1および2の情報「ああ」および「いい」がそのまま設定される。

【0044】また、個別格納レコードとMASKテーブル1（共通フィールドがHIGH-VALUE値で、個別フィールドが「NULL値」のテーブル）とで論理積をとった結果1の全フィールドが「NULL値」でない場合、共通フィールドの情報が共通格納レコードの情報と不一致であると判断できるので、図8のときと同様の2分割により不一致の範囲が共通フィールド1に絞り込まれる。図11は、個別格納レコード④と共通格納レコード①から入力レコードcを復元する場合を示している。

【0045】図11において、一致している範囲の情報は、図10の場合と同様に、個別格納レコードと共通格納レコードとの論理和をとることによって復元される。すなわち、個別格納レコード④の共通フィールド2と共通格納レコード①の個別フィールド1および2は「NULL値」であるので、復元後のレコードc'は、共通フィールド2に共通格納レコード①の共通フィールド2の情報「CC」が、個別フィールド1および2には個別格納レコード④の個別フィールド1および2の情報「おお」および「かか」が設定される。また、最終的に絞り込まれた不一致のフィールド（すなわち「共通フィールド1」）の情報は、個別格納レコード④の情報「BB」

を優先して設定することによって、復元後レコードc'が復元される。

【0046】

【発明の効果】以上説明したように、本発明のデータベースの重複項目格納方法によれば、同一キー内（枝番などで複数のレコードが存在する）の複数レコード間で、重複している情報を、共通格納レコードとして集約（個別項目情報は「NULL値」）するとともに、個別格納レコードは共通格納レコードと逆の重複しない情報を設定（重複項目情報は「NULL値」）し作成しているため、データベースに格納する場合、データベースの標準機能である「連続文字圧縮機能」が有効に働き、ディスク容量が大幅に削減できるという顕著な効果がある。

【0047】また、共通格納レコードに設定するフィールドと、個別格納フィールドに設定するフィールドの識別は、MASKテーブルとの比較によって実現させているため、共通情報、個別情報に変更があっても、MASKテーブルの修正で対応できるため、ユーザプログラムの修正は不要となり、容易に変更できるという効果がある。さらに、格納レコードの復元は、共通格納レコードと個別格納レコードの論理和により実現しているため、1項目単位に比較する方法に比較して処理効率を大幅に向上させる効果がある。

【図面の簡単な説明】

【図1】本発明の実施例に係るユーザプログラム、DBアクセス制御ルーチン、データベース管理システム（DBMS）、およびリレーショナル・データベースの関連を示す概念図である。

【図2】データベースへ格納するためのデータ作成イメージを表わす図である。

【図3】データベースに格納されている格納レコードからの復元イメージを表わす図である。

【図4】最初に入力レコードから格納レコード（共通格納レコード、個別格納レコード）を作成する方法を示す図である。

【図5】2番目以降の入力レコードから格納レコード（個別格納レコード）を作成する方法を示す図である（共通フィールドが同一の場合）。

【図6】2番目以降の入力レコードから格納レコード（個別格納レコード）を作成する方法を示す図である（共通フィールドが異なる場合）。

【図7】共通格納レコードと個別格納レコードの共通フィールドの情報の一致／不一致の判定方法を示す図である（一致の場合）。

【図8】共通格納レコードと個別格納レコードの共通フィールドの情報の一致／不一致の判定方法および不一致の共通フィールドの見つけ出す方法を示す図である（不一致の場合）。

【図9】格納レコードから入力レコードへ復元する場合、個別格納レコードの共通フィールドの情報が共通格

納レコードの共通フィールドの情報と一致しているか否かを見つけて出す方法を示す図である。

【図10】格納レコードから入力レコードへ復元する場合、個別格納レコードの共通フィールドの情報が共通格納レコードの共通フィールドの情報と一致しているときの復元方法を示す図である。

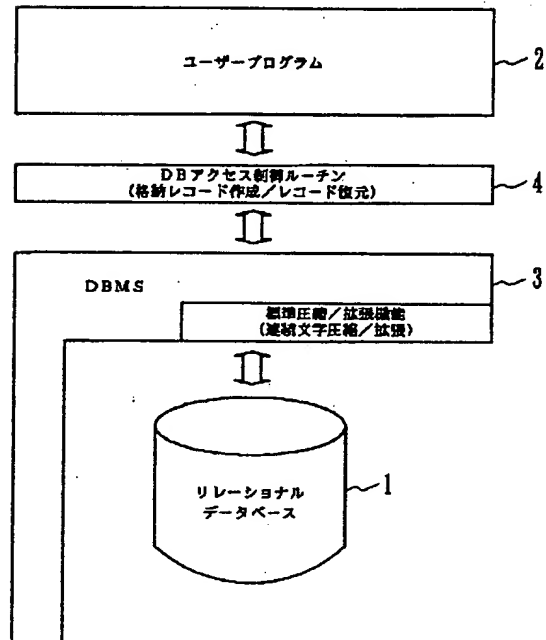
【図11】格納レコードから入力レコードへ復元する場

合、個別格納レコードの共通フィールドの情報が共通格納レコードの共通フィールドの情報と不一致のときの復元方法を示す図である。

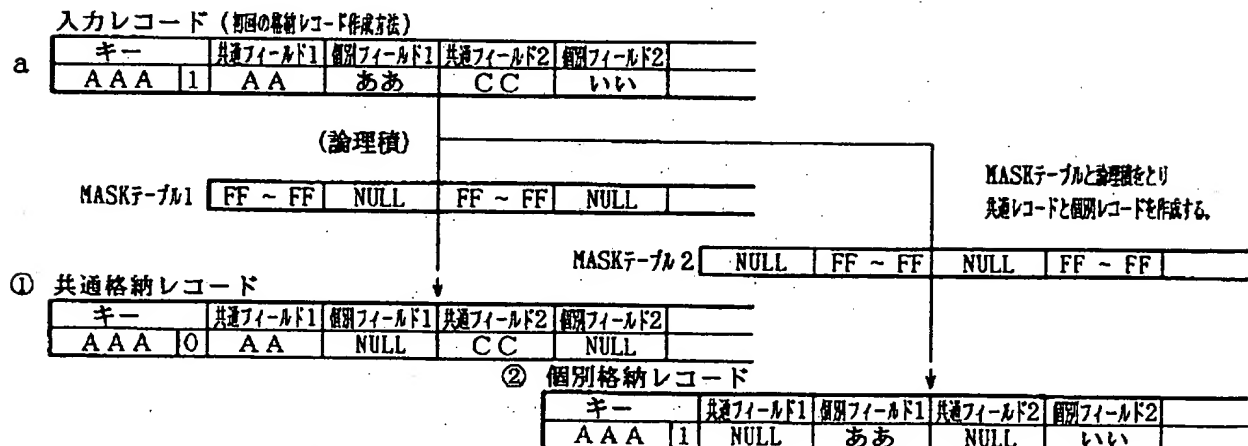
【符号の説明】

1：リレーショナル・データベース、2：ユーザプログラム、3：データベース管理システム（DBMS）、4：DBアクセス制御ルーチン

【図1】



【図4】



【図2】

入力レコードイメージ

a	キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
	AAA	1	AA	ああ	CC
b	キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
	AAA	2	AA	うう	CC
c	キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
	AAA	3	BB	おお	CC
d	キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
	AAA	4	AA	きき	DD

入力レコードより、共通格納レコードと個別格納レコードを作成する。

個別格納レコードを作成する。

個別格納レコードを作成する。

個別格納レコードを作成する。

格納レコードイメージ

① 共通格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	0	AA	CC	NULL

② 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	1	NULL	ああ	NULL

③ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	2	NULL	うう	NULL

④ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	おお	NULL

⑤ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	4	NULL	きき	DD

分離

【図5】

入力レコード（2番目以降の格納レコード作成方法）

b	キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
	AAA	2	AA	うう	CC

〈排他的論理和〉

① 共通格納レコード	AA	NULL	CC	NULL
------------	----	------	----	------

1番目で作成した共通レコードとで排他的論理和をとり、個別レコードを作成する。

③ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	2	NULL	うう	NULL

【図3】

格納レコードイメージ

① 共通格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	0	AA	NULL	CC

② 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	1	NULL	ああ	NULL

③ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	2	NULL	うう	ええ

④ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	おお	NULL

⑤ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	4	NULL	きき	DD

復元後レコードイメージ (入力レコードと同じ)

a'

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	1	AA	ああ	CC

b'

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	2	AA	うう	CC

c'

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	おお	CC

d'

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	4	AA	きき	DD

共通格納レコードと
個別格納レコードより
作成する。

【図6】

入力レコード (2番目以降で共通フィールドの内容が異なる格納レコード作成方法)

c

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	おお	CC

① 共通格納レコード

AA	NULL	CC	NULL
----	------	----	------

(排他的論理和)

1番目で作成した共通レコードと排他的論理和をとり、
個別レコードを作成する。
但し、共通フィールドの内容が異なる場合、入力レコードの内容を優先する。

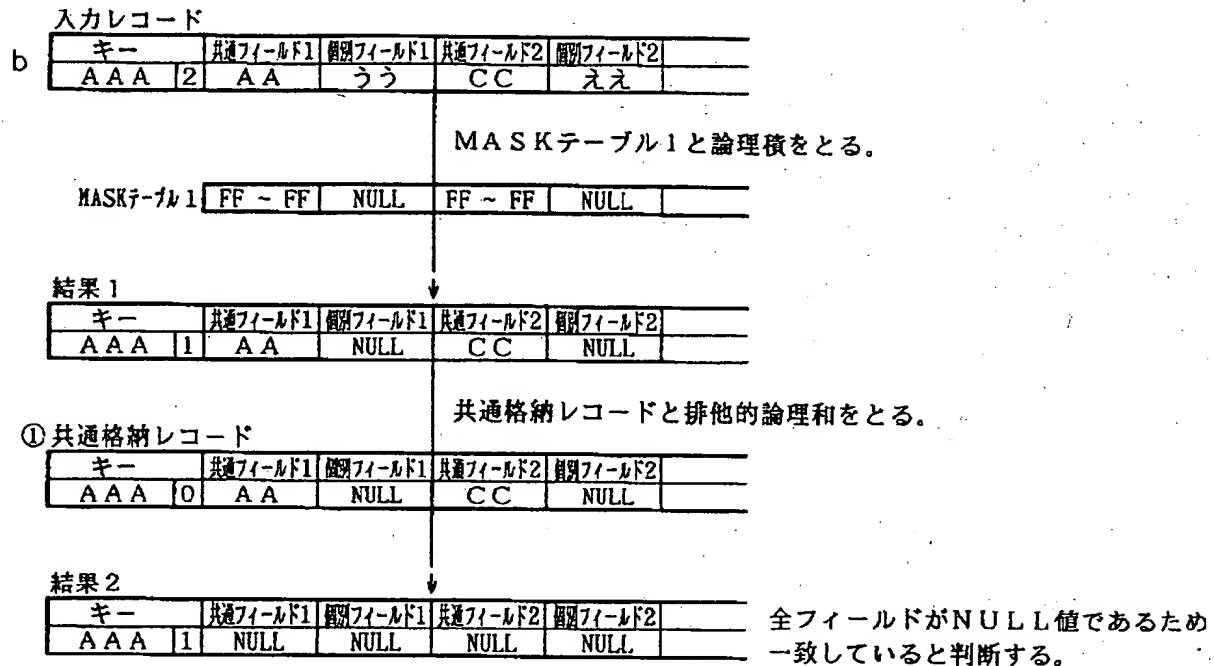
④ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	おお	NULL

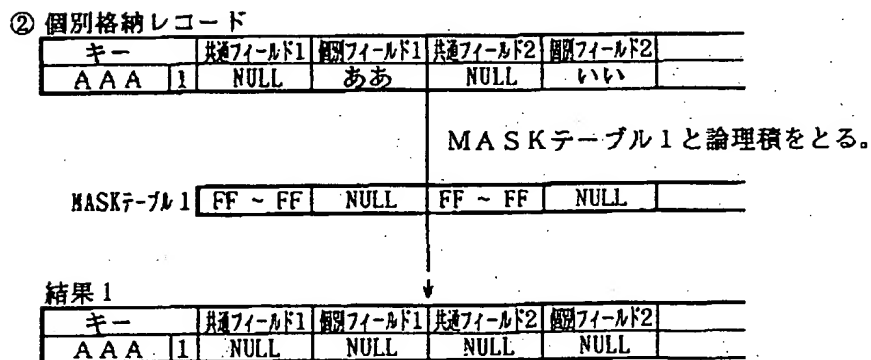
共通項目であるが、共通格納レコードと内容が異なる場合、NULL値でなく
入力値を設定する。

【図7】

共通フィールドの情報が一致している場合

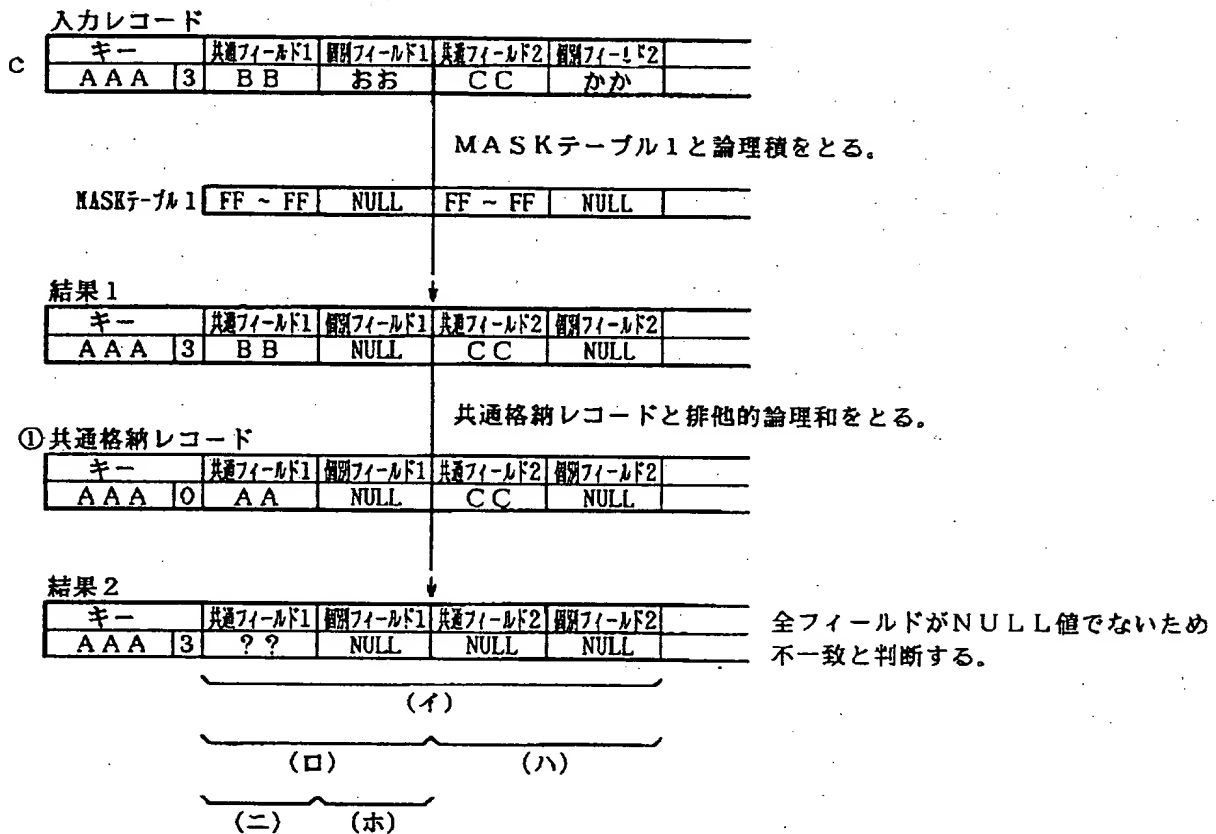


【図9】



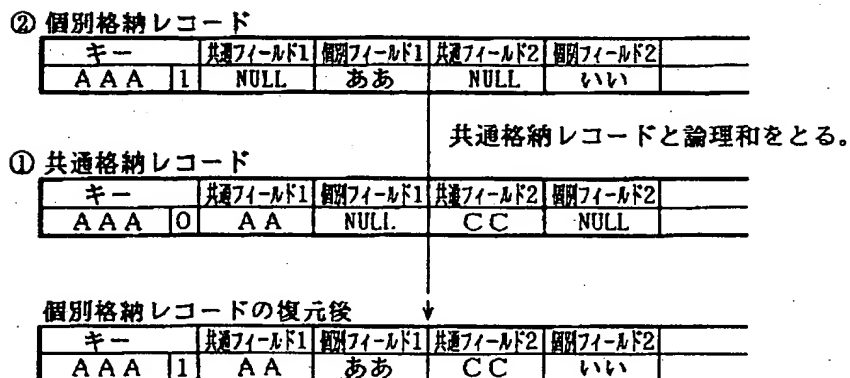
【図8】

共通フィールドの情報が不一致の場合



【図10】

結果1の値が、ALL NULL値の場合



【図11】

結果1の値が、ALL NULL値でない場合

④ 個別格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	お	お

共通格納レコードと論理和をとる。

① 共通格納レコード

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	0	AA	NULL	CC

不一致フィールドは個別格納レコードの値を優先する。

個別格納レコードの復元後

キー	共通フィールド1	個別フィールド1	共通フィールド2	個別フィールド2
AAA	3	BB	お	お